

Une approche basée sur l'ASP pour la représentation des réseaux booléens et la détection des attracteurs : application aux réseaux de gènes

Tarek KHALED et Belaïd Benhamou

BIOSS-IA

24 Novembre 2020



- 1 Introduction
- 2 Réseaux Booléens
- 3 Généralités sur les programmes logiques et l'ASP
- 4 Recherche d'attracteurs dans les réseaux booléens **synchrones et asynchrones**
- 5 Recherche d'attracteurs dans les réseaux booléens **circulaires asynchrones**
- 6 Conclusion

- Les réseaux booléens ont été adoptés pour raisonner sur les réseaux de gènes car ils nécessitent peu de paramètres.
 - Nous manquons d'informations quantitatives sur les constantes de liaison ou les paramètres cinétiques
 - Nous pouvons compter que sur une description qualitative du type "A active (ou inhibe) B"
- Un objectif central de l'analyse des réseaux Booléens est la détermination des attracteurs

- La dynamique d'un réseau booléen peut converger vers un ensemble de configuration
- Les attracteurs peuvent être singleton ou cyclique
- Les attracteurs cycliques peuvent correspondre aux cycles cellulaires (croissance)
- Les attracteurs singleton peuvent correspondre à des états différenciés ou d'apoptose

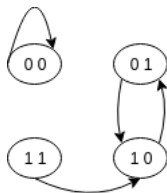
Definition

- Soit $X = \{0, 1\}^n$ l'espace de configuration d'un réseau booléen et $f : X \rightarrow X$ sa fonction de transition globale associée.
- Le graphe de transition représentant la dynamique de f est le graphe orienté $TG(f) = (X, T(f))$ où l'ensemble des sommets est l'espace de configuration X et l'ensemble des arcs est
- $T(f) = \{(x, y) \in X^2 \mid x \neq y, x = (x_1, \dots, x_i, \dots, x_n), y = (f_1(x), \dots, f_i(x), \dots, f_n(x))\}$

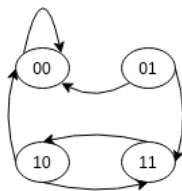
- La dynamique du réseau Booléen est décrite par un graphe de transition TG défini par sa **fonction de transition** f et un **mode de mise à jour**.
- $f : X \rightarrow X$ tel que $x = (x_1, \dots, x_n) \mapsto f(x) = (f_1(x), \dots, f_n(x))$.
- Le mode **synchrone** veut dire que tous les composants d'une configuration $x = (x_1, \dots, x_n)$ sont mis à jour au **même moment**.
- Le mode **asynchrone** veut dire qu'un composant est mis à jour à chaque fois.

- Un cycle de TG est une sequence de configurations (x^1, \dots, x^r, x^1)
- Une sequence $(x^1, x^2, \dots, x^r, x^1)$ forme un cycle stable de TG lorsque $\forall t < r, x^{t+1}$ est l'unique successeur de x^t et x^1 est l'unique successeur de x^r .
- $x = (x_1, \dots, x_n)$ est une configuration stable de TG lorsque $\forall x_i \in V, x_i = f_i(x)$.

- **Exemple** : Soit $X = \{1, 2\}$ et la fonction de transition f définie par :
 $f(x_1, x_2) = (x_2, x_1 \wedge \neg x_2)$.



Graphe de transition
Synchrone



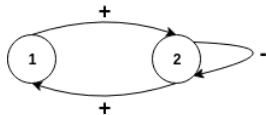
Graphe de transition
Asynchrone

- Les données biologiques sont des corrélations entre les gènes
- Représenter par un graphe d'interaction IG
- Chaque nœud du graphe d'interaction représente un gène.

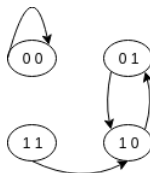
Definition

Un graphe d'interaction est un graphe orienté signé $IG = (V, I)$ où $V = \{1, \dots, n\}$ est l'ensemble des sommets et $I \subseteq V \times \{+, -\} \times V$ est l'ensemble des arcs signés.

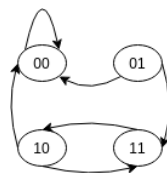
- **Exemple** : Soit $X = \{1, 2\}$ et la fonction de transition f définie par :
 $f(x_1, x_2) = (x_2, x_1 \wedge \neg x_2)$.



Graphe D'interaction



Graphe TG
Synchrone

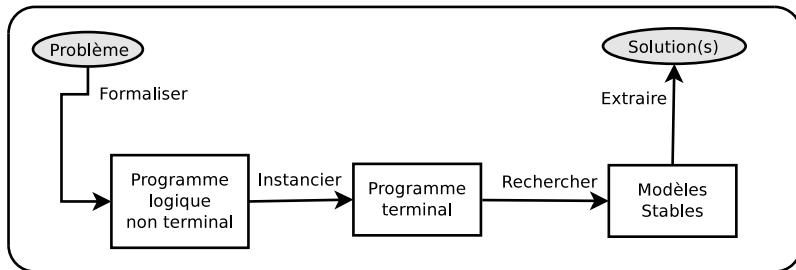


Graphe TG
Asynchrone

Généralités sur les programmes logiques et l'ASP

Contexte général

- Un paradigme de programmation déclarative basé sur la logique.
- Les connaissances sont codées par un programme logique.
- Le sens d'un programme logique est capturé par la sémantique utilisée.



Généralités sur les programmes logiques et l'ASP

- Un programme logique π est composé par un ensemble de règles de la forme $r : \text{tête}(r) \leftarrow \text{corps}(r)$.
- Un **programme positif** ne contient pas de négation par échec ou de négation classique ($A_0 \leftarrow A_1, A_2, \dots, A_m$) où A_i est un atome.
- Un **programme général** π est un ensemble de règles de la forme : $r = A_0 \leftarrow A_1, A_2, \dots, A_m, \text{not } A_{m+1}, \dots, \text{not } A_n$ où A_i est un atome.
- Un **programme étendu** π est un ensemble de règles de la forme : $r = L_0 \leftarrow L_1, L_2, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_n$ où L_i est un littéral (A_i ou $\neg A_i$)

Généralités sur les programmes logiques et l'ASP

La sémantique des modèles stables

- **Programme positif** : admet un seul modèle stable qui son modèle minimal de Herbrand.
- **Programme général(réduit)** : Le réduit d'un programme général π par rapport à un ensemble d'atomes X est le programme positif π^X obtenu à partir de π en supprimant :
 - chaque instance de règle ayant un atome *not* A_i tel que $A_i \in X$
 - tous les atomes *not* A_j tels que $A_j \notin X$, dans les règles restantes.
- **Programme général (modèles stables)** :
 - X est un modèle stable de programme général π ssi X est identique au modèle minimal de Herbrand de π^X . $X = Cn(\pi^X)$
 - La sémantique des modèles stables est basé sur l'hypothèse du monde clos.

Généralités sur les programmes logiques et l'ASP

La nouvelle sémantique [Benhamou et Siegel, 2012]

- Elle est basée sur un langage propositionnel classique L ayant deux types de variables :
 - $V = \{A_i : A_i \in L\}$
 - $nV = \{not A_i : not A_i \in L\}$
- Pour chaque variable $A_i \in V$, il existe une variable correspondante $not A_i \in nV$ (la négation par échec de A_i).
- Un lien entre les deux types de variables est exprimé par l'ensemble de clauses
$$ME = \{(\neg A_i \vee \neg not A_i) : A_i \in V\}.$$

Généralités sur les programmes logiques et l'ASP

La nouvelle sémantique [Benhamou et Siegel, 2012]

- $\pi = \bigcup_{r \in \pi} \{A_0 \leftarrow A_1, A_2, \dots, A_m, \text{not } A_{m+1}, \dots, \text{not } A_n\}$ est un **programme logique général** alors sa forme Horn clause est :
 - $$\text{HC}(\pi) = \left\{ \bigcup_{r \in \pi} (A_0 \vee \neg A_1 \vee \dots \vee \neg A_m \vee \neg \text{not } A_{m+1}, \dots, \neg \text{not } A_n) \right. \\ \left. \bigcup_{A_i \in V} (\neg A_i \vee \neg \text{not } A_i) \right\}.$$
 - L'ensemble Strong Backdoor de π est :
$$STB = \{\text{not } A_i : \exists r \in \pi, A_i \in \text{body}^-(r)\} \subseteq nV$$

Généralités sur les programmes logiques et l'ASP

La nouvelle sémantique [Benhamou et Siegel, 2012]

Definition

Soit $HC(\pi)$ le codage CNF d'un programme logique π , STB son strong backdoor et un sous-ensemble $S' \subseteq STB$, l'ensemble $E = HC(\pi) \cup S'$ de clauses est alors une extension de $(HC(\pi), STB)$ si les conditions suivantes sont vérifiées :

- 1 E est consistant,
- 2 $\forall \text{not } A_i \in STB - S', E \cup \{\text{not } A_i\}$ est inconsistent.

Proposition

Soit π un programme logique et STB son strong backdoor. Si $HC(\pi)$ est consistant, alors il existe au moins une extension de la pair $(HC(\pi), STB)$.

Généralités sur les programmes logiques et l'ASP

La nouvelle sémantique [Benhamou et Siegel, 2012]

Théorème

Si X est un modèle stable d'un programme logique π , alors il existe une extension E de $(HC(\pi), STB)$ telle que $X = \{A_i \in V : E \models A_i\}$. D'autre part, E vérifie la condition dite discriminante :

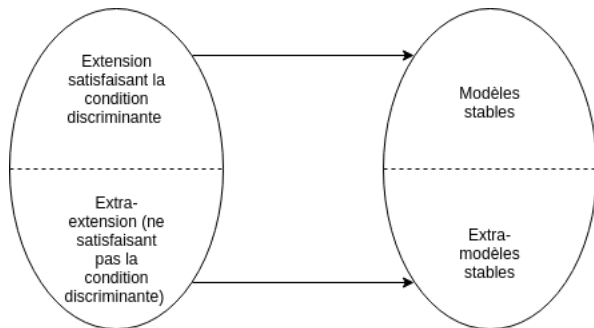
$$(\forall A_i \in V, E \models \neg \text{not } A_i \Rightarrow E \models A_i).$$

Théorème

Si E est une extension de $(HC(\pi), STB)$, qui vérifie la condition discriminante $(\forall A_i \in V, E \models \neg \text{not } A_i \Rightarrow E \models A_i)$, alors $X = \{A_i : E \models A_i\}$ est un modèle stable de π .

Généralités sur les programmes logiques et l'ASP

La nouvelle sémantique [Benhamou et Siegel, 2012]



Extension de la nouvelle sémantique aux programmes logiques étendus

- Un programme étendu π est réduit vers un programme général π^+ .
- Tous les littéraux $\neg L$ dans π sont remplacés par L' .
- Nous rajoutons des contraintes d'intégrités $\leftarrow L, L'$.
- Chaque modèle stable du programme général π^+ correspond à un ensemble réponse du programme étendu π et vice versa.

Une nouvelle méthode pour la recherche des modèles stables

- La méthode est un processus d'énumération type **DPLL**.
- Cette méthode calcule toutes les extensions de $(HC(\pi), STB)$.
- La recherche d'extension se fait par l'ajout progressive de littéraux *not* A_i du STB .
- Si on s'intéresse aux modèles stables
 - Il suffit de retenir que les extensions qui vérifient la condition discriminante

Une nouvelle méthode pour la recherche des modèles stables

Étapes principales de la méthode

- La recherche de modèle stable alterne des phases de propagation unitaire déterministes et des phases de points de choix non déterministes
- Une extension est trouvée quand :
 - Toutes les clauses sont satisfaites
 - Tous les littéraux du STB ont été affectés sans falsifier aucune clause.
- La méthode complète l'interprétation courante par
 - l'assignation à vrai de l'ensemble des variables *not* A_i restant du *STB*.
 - l'assignation à faux de toutes les autres variables A_i non encore affectés (hypothèse du monde clos).

Recherche d'attracteurs dans les réseaux booléens synchrones et asynchrones

[Khaled and Benhamou in ICLP 2019]

- Exprimer les réseaux booléens et simuler la dynamique
- Énumération de tous les attracteurs des réseaux booléens
 - synchrones
 - asynchrones
- L'identification des attracteurs revient à énumérer les modèles stables
- La vérification de l'existence des attracteurs dans une séquence de configurations

Recherche d'attracteurs dans les réseaux booléens

Modélisation du graphe d'interactions

- Initialisation : $v_i(0) \leftarrow \text{not } \neg v_i(0).$ $\neg v_i(0) \leftarrow \text{not } v_i(0).$
- Arc positif : $v_j(t+1) \leftarrow v_i(t)$ $\neg v_j(t+1) \leftarrow \neg v_i(t)$
- Arc négatif : $v_j(t+1) \leftarrow \neg v_i(t)$ $\neg v_j(t+1) \leftarrow v_i(t)$
- Règles d'inertie :
 - $v_i(t+1) \leftarrow v_i(t), \text{ not } \neg v_i(t+1)$
 - $\neg v_i(t+1) \leftarrow \neg v_i(t), \text{ not } v_i(t+1)$

Recherche d'attracteurs dans les réseaux booléens

Modélisation synchrone

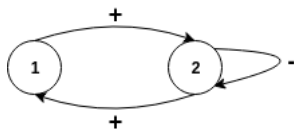
- La fonction de transition locale $f_j : v_j(t+1) = f_j(v(t)) = \bigvee_{i=1}^{l_j} m_i^j$.
- $DNF(\neg f_j(v(t))) = \neg(\bigvee_{i=1}^{l_j} m_i^j) = \bigvee_{i=1}^{r_j} m_i'^j$ la forme DNF de $\neg f_j(v(t))$.
- Les formules m_i^j et $m_i'^j$ sont des **conjonctions de littéraux**
- m_i^j et $m_i'^j$ représentent les interactions positive/négative des gènes agissant sur $v_j(t)$.

Recherche d'attracteurs dans les réseaux booléens

Modélisation synchrone

• Exemple :

$$f(x_1, x_2) = (x_2, x_1 \wedge \neg x_2)$$



$$\begin{aligned}\pi(IG) = & \{1(t+1) \leftarrow 2(t); \\ & 2(t+1) \leftarrow 1(t), \neg 2(t) \\ & \neg 1(t+1) \leftarrow \neg 2(t); \\ & \neg 2(t+1) \leftarrow \neg 1(t); \\ & \neg 2(t+1) \leftarrow 2(t)\}\end{aligned}$$

Recherche d'attracteurs dans les réseaux booléens

Modélisation asynchrone

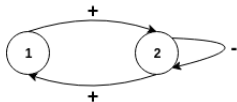
- $Change(v_i, t) \leftarrow v_i(t+1), \neg v_i(t)$
 $Change(v_i, t) \leftarrow \neg v_i(t+1), v_i(t)$
- $\{\{Block(v_k, t)\} \leftarrow Change(v_i, t), not Block(v_i, t) \mid \forall k \in \{1, \dots, n\} \setminus \{i\}\}$
- $\{v_j(t+1) \leftarrow m_i^j(t), not Block(v_j, t) \mid 1 \leq i \leq l_j\}, j \in \{1, \dots, n\}$
- $\{\neg v_j(t+1) \leftarrow m_i^j(t), not Block(v_j, t) \mid 1 \leq i \leq r_j\}, j \in \{1, \dots, n\}$

Recherche d'attracteurs dans les réseaux booléens

Modélisation asynchrone

• Exemple :

$$f(x_1, x_2) = (x_2, x_1 \wedge \neg x_2)$$



$$\pi(IG) =$$

$$\{Block(1, t) \leftarrow Change(2, t), not\ Block(2, t)\}$$

$$Block(2, t) \leftarrow Change(1, t), not\ Block(1, t)\}$$

$$\{1(t+1) \leftarrow 2(t), not\ Block(1, t);$$

$$2(t+1) \leftarrow 1(t), \neg 2(t), not\ Block(2, t)$$

$$\neg 1(t+1) \leftarrow \neg 2(t), not\ Block(1, t);$$

$$\neg 2(t+1) \leftarrow \neg 1(t), not\ Block(2, t);$$

$$\neg 2(t+1) \leftarrow 2(t), not\ Block(2, t)\}$$

Recherche d'attracteurs dans les réseaux booléens

La recherche d'attracteurs

Proposition

- Soit P_{IG} le programme logique représentant le graphe d'interaction IG ayant une fonction de transition globale f et $TG(f)$ le graphe de transition correspondant.
- Un tuple $x = (x^0, \dots, x^t)$ est une séquence de configurations de $TG(f)$, si seulement si
- $I = \{(v_1(0), \dots, v_n(0)), \dots, (v_1(t), \dots, v_n(t))\}$ est un ensemble réponses de P_{IG} tel que l'ensemble des littéraux $(v_1(i), \dots, v_n(i))$ fixé à l'étape $i \in \{0, \dots, t\}$ correspond à l'état des gènes de la configuration $x^i \subseteq x$ défini à l'étape i dans le graphe de transition $TG(f)$.

Recherche d'attracteurs dans les réseaux booléens

La recherche d'attracteurs

- Énumérer toutes les configurations initiales possibles et exécuter une simulation à partir de chacune d'elles.
- Une énumération explicite de toutes les configurations est fastidieuse pour les grands réseaux.
- On rajoute des contraintes pour interdire de rechercher les cycles déjà trouvés
- Si pour une longueur de séquence donnée, nous ne trouvons aucun cycle, l'algorithme double la taille.
- L'algorithme s'arrête lorsque aucune séquence de configurations n'est trouvée.

Recherche d'attracteurs dans les réseaux booléens

La recherche d'attracteurs

- Déterminer la présence d'un cycle en vérifiant si le dernier état survient au moins deux fois.
- Détection d'une configuration stable en vérifiant que la dernière configuration a comme successeur elle-même.
- Dans le graphe de transition synchrone, les configurations ont un unique successeur.
- lorsque une séquence de configurations entre dans un cycle, il ne le quitte jamais.

Recherche d'attracteurs dans les réseaux booléens

La recherche d'attracteurs

- Dans le graphe asynchrone, les configurations peuvent avoir plusieurs transitions.
- Nous devons vérifier si une configuration peut évoluer vers une configuration en dehors du cycle.
- Nous devons vérifier que pour chaque $\{(v_1(i), \dots, v_n(i))\}$ correspondant à x^i
 - Il n'y pas un nouveau successeur $\{(v_1(i+1), \dots, v_n(i+1))\}$ correspondant à x^{i+1}
 - Ce nouveau successeur doit être différent de celui dans le cycle
- Nous essayons de produire une configuration différente à chaque point de choix *not* $Block(v_i, t)$.

Recherche d'attracteurs dans les réseaux booléens

Les résultats expérimentaux

Réseau	Gènes	Attracteurs	Mode de mis à jour	Temps(Sec)
Yeast cell cycle	11	6	Synchrone	2,21
	11	6	Asynchrone	0,56
Fission Yeast	10	11	Synchrone	1,82
	10	12	Asynchrone	0,5
Th cell differentiation	23	2	Synchrone	0,37
	23	2	Asynchrone	0,43

Recherche des attracteurs dans les réseaux booléens circulaires

[Khaled and Benhamou in LPAR-23(2020)]

- Nous portons une attention particulière aux réseaux Booléens circulaires.
- Nous allons nous concentrer sur le mode asynchrone.
- La détection des attracteurs se fait sans passer par la simulation des réseaux Booléens.
- Il existe certaines propriétés communes entre les circuits et la nouvelle sémantique notamment les extra-extensions (extra-modèles).

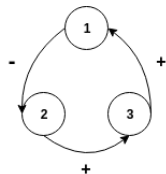
Definition

- *Un graphe d'interaction circulaire $IG = (V, I)$ de taille k est une séquence $C = (i_1, i_2, \dots, i_k, i_1)$ de sommets de V telle que $I = \{(i_j, \{+, -\}, i_{j+1}) \mid \forall j \in \{1, \dots, k-1\}\} \cup \{(i_k, \{+, -\}, i_1)\}$*
- *Si le nombre d'arcs étiquetés par le signe "-" (arcs négative) est paire (resp. impaire), alors le circuit C est positif (resp. négatif).*

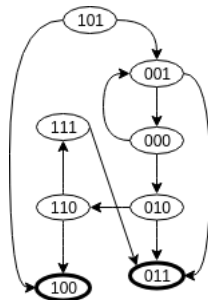
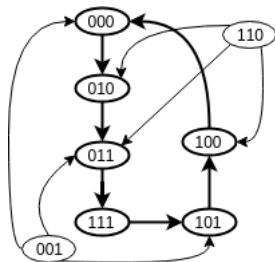
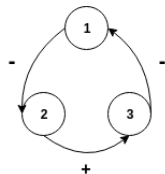
Détection d'attracteurs dans les réseaux Booléens circulaires

Réseaux Booléens circulaires

$$f(x_1, x_2, x_3) = (x_3, \neg x_1, x_2)$$



$$g(x_1, x_2, x_3) = (\neg x_3, \neg x_1, x_2)$$



Definition

- i signifie que le gène i est actif dans la cellule
- $\neg i$ signifie que le gène i n'est pas actif.
- $\text{not } \neg i$ (resp. $\neg \text{not } \neg i$) signifie que la cellule donne (resp. ne donne pas) l'autorisation d'activer i .
- $\text{not } i$ (resp. $\neg \text{not } i$) signifie que la cellule donne (resp. ne donne pas) le droit de désactiver i .

Détection d'attracteurs dans les réseaux Booléens circulaires

Représentation logique

Definition

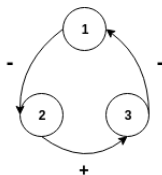
La traduction de IG en un programme logique P_{IG} se fait en traduisant chaque arc de IG en une paire de règles. Plus précisément :

- *Un arc positif $(i, +, j)$ est traduit en : $j \leftarrow \text{not } \neg i, \neg j \leftarrow \text{not } i$*
- *Un arc négatif $(i, -, j)$ est traduit en : $j \leftarrow \text{not } i, \neg j \leftarrow \text{not } \neg i$*

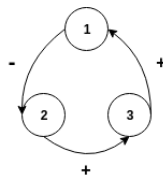
Détection d'attracteurs dans les réseaux Booléens circulaires

Représentation logique

Exemple :



$$P_{IG_p} = \{ \\ 2 \leftarrow \text{not } 1, \\ \neg 2 \leftarrow \text{not } \neg 1, \\ 3 \leftarrow \text{not } \neg 2, \\ \neg 3 \leftarrow \text{not } 2, \\ 1 \leftarrow \text{not } 3, \\ \neg 1 \leftarrow \text{not } \neg 3 \}$$



$$P_{IG_n} = \{ \\ 2 \leftarrow \text{not } 1, \\ \neg 2 \leftarrow \text{not } \neg 1, \\ 3 \leftarrow \text{not } \neg 2, \\ \neg 3 \leftarrow \text{not } 2, \\ 1 \leftarrow \text{not } \neg 3, \\ \neg 1 \leftarrow \text{not } 3 \}.$$

Definition

Soit IG un graphe d'interaction ayant l'ensemble de sommets $V = \{1, \dots, n\}$ et E une extension de $HC(P_{IG})$ obtenue en ajoutant à $HC(P_{IG})$ un ensemble maximal consistant de $\{\text{not } i\}$, avec $i \in \{1, \dots, n, \neg 1, \dots, \neg n\}$. Nous avons :

- 1 E est complet si pour tous les $i \in V$, $\text{not } \neg i \in E$ ou $\text{not } i \in E$.
- 2 i est libre dans E si $i \notin E$ et $\neg i \notin E$. Autrement, i est fixé.
- 3 Le degré de liberté de E , noté $\text{deg}(E)$, est le nombre de sommets libres qui le compose.
- 4 Le miroir de $E = HC(P_{IG}) \cup \{\text{not } i \mid i \in V\}$, noté $\text{mir}(E)$, est défini comme $\text{mir}(E) = HC(P_{IG}) \cup \{\text{not } \neg i \mid i \in V\}$.

Détection d'attracteurs dans les réseaux Booléens circulaires

Représentation logique

Proposition

Soit $HC(P_{IG})$ un programme logique représentant le graphe d'interaction IG . Alors, Pour chaque i , $\neg(\text{not } \neg i \wedge \text{not } i)$ est vrai.

Proposition

Soit IG un graph d'interaction, E est une extension de $HC(P_{IG})$. Si chaque nœud d'un graphe d'interaction IG a au moins un arc entrant, alors toute extension complète de $HC(P_{IG})$ est de degré 0.

Proposition

Soit IG un graphe d'interaction, si chaque sommet de IG a au moins un arc entrant, alors toute extension complète de $HC(P_{IG})$ correspond à un ensemble réponse de $HC(P_{IG})$.

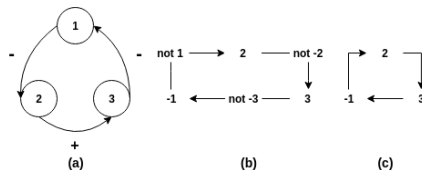
Détection d'attracteurs dans les réseaux Booléens circulaires

Représentation logique

Proposition

Si le graphe d'interaction IG est un circuit positif de taille n alors $HC(P_{IG})$ a 2 d'extension de degré 0

Exemple :



$$\begin{aligned}
 HC(P'_{IG_p}) = \{ & \\
 & 2 \vee \neg \text{not } 1, \\
 & 2' \vee \neg \text{not } 1', \\
 & 3 \vee \neg \text{not } 2', \\
 & 3' \vee \neg \text{not } 2, \\
 & 1 \vee \neg \text{not } 3, \\
 & 1' \vee \neg \text{not } 3' \\
 & \neg \text{not } i' \vee \neg \text{not } i \\
 & i \in \{1, 2, 3\} \}
 \end{aligned}$$

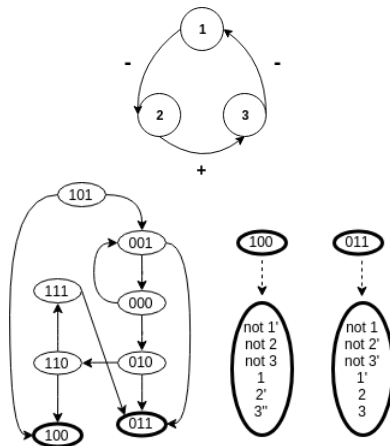
$$E_1 = HC(P'_{IG_p}) \cup \{\text{not } 1, \text{not } 2', \text{not } 3'\}$$

$$E_2 = HC(P'_{IG_p}) \cup \{\text{not } 1', \text{not } 2, \text{not } 3\}.$$

Les modèles stables de P'_{IG_p} sont $M'_1 = \{1', 2, 3\}$ et $M'_2 = \{1, 2', 3'\}$ correspondant aux ensembles réponses $M_1 = \{\neg 1, 2, 3\}$ et $M_2 = \{1, \neg 2, \neg 3\}$ de P_{IG_p}

Détection d'attracteurs dans les réseaux Booléens circulaires

Relation entre la représentation logique et le graphe de transition



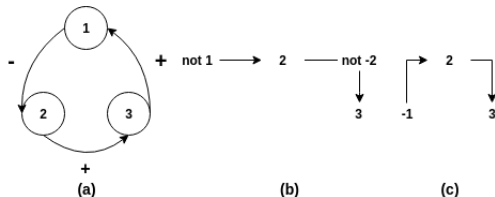
Détection d'attracteurs dans les réseaux Booléens circulaires

Représentation logique

Proposition

Si le graphe d'interaction IG est un circuit négatif de taille n alors $HC(P_{IG})$ a $2n$ extra-extensions (extra-modèles) de degré 1.

Exemple :



$$HC(P'_{IG_n}) = \{$$

$$2 \vee \neg \text{not } 1,$$

$$2' \vee \neg \text{not } 1',$$

$$3 \vee \neg \text{not } 2',$$

$$3' \vee \neg \text{not } 2,$$

$$1 \vee \neg \text{not } 3',$$

$$1' \vee \neg \text{not } 3$$

$$\neg \text{not } i' \vee \neg \text{not } i$$

$$i \in \{1, 2, 3\}\}.$$

$$E'_1 = HC(P'_{IG_n}) \cup \{\text{not } 1, \text{not } 2'\}, E'_1 \models \{\text{not } 1, 2, \text{not } 2', 3\}$$

$$E'_2 = HC(P'_{IG_n}) \cup \{\text{not } 1, \text{not } 3\}, E'_2 \models \{\text{not } 1, 2, \text{not } 3, 1'\}$$

$$E'_3 = HC(P'_{IG_n}) \cup \{\text{not } 1', \text{not } 3'\}, E'_3 \models \{\text{not } 1', 2', \text{not } 3', 1\}$$

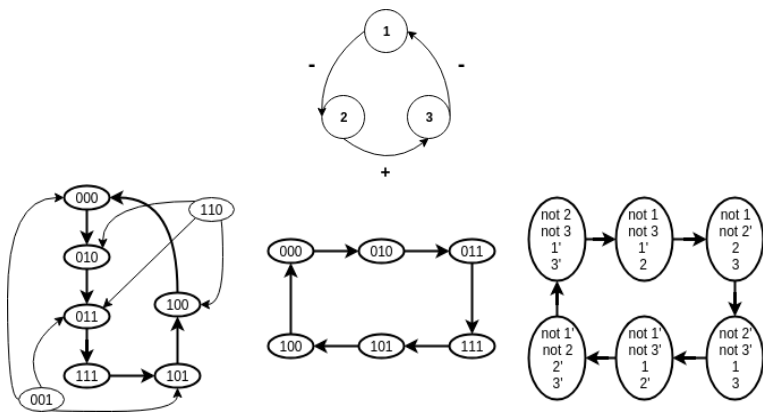
$$E'_4 = HC(P'_{IG_n}) \cup \{\text{not } 1', \text{not } 2\}, E'_4 \models \{\text{not } 1', 3', \text{not } 2, 3'\}$$

$$E'_5 = HC(P'_{IG_n}) \cup \{\text{not } 2', \text{not } 3'\}, E'_5 \models \{\text{not } 2', 3, \text{not } 3', 1\}$$

$$E'_6 = HC(P'_{IG_n}) \cup \{\text{not } 2, \text{not } 3\}, E'_6 \models \{\text{not } 2, 3', \text{not } 3, 1'\}$$

Détection d'attracteurs dans les réseaux Booléens circulaires

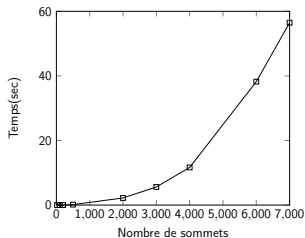
Relation entre la représentation logique et le graphe de transition



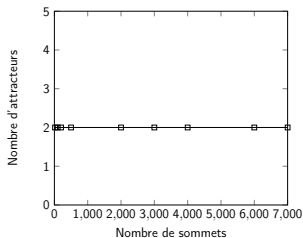
Détection d'attracteurs dans les réseaux Booléens circulaires

Validation Empirique

Le temps d'exécution obtenu sur les graphes circulaires positifs



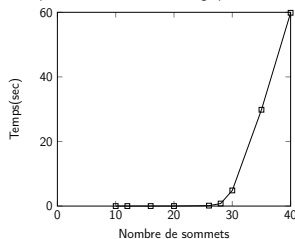
Le nombre de configurations stables sur les graphes circulaires positifs



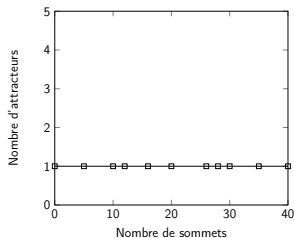
Détection d'attracteurs dans les réseaux Booléens circulaires

Validation Empirique

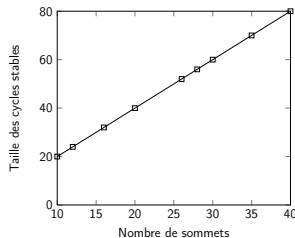
Le temps d'exécution obtenu sur les graphes circulaires négatifs



Le nombre de configurations stables sur les graphes circulaires négatifs



La taille des cycles stables sur les graphes circulaires négatifs



- Détection de configurations et de cycles stables dans les réseaux de régulation génétiques.
 - **Cas général** : Simulation de la dynamique des réseaux booléens, puis vérification de l'existence des attracteurs sur des réseaux de gènes réels.
 - **Cas particulier** : Détection des attracteurs dans les réseaux booléens circulaires (pas de simulation).
- Détection des attracteurs d'une manière totalement déclarative dans le cas général.
- Prendre en compte d'autres modes de mise à jour dans le cas de détection d'attracteurs dans les réseaux Booléens circulaires.

Merci pour votre attention